UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

SEGMENTACIÓN DE OBJETOS EN VÍDEOS NO CONTROLADOS

Ariana Vicario Arroyo
Tutor: Diego Ortego Hernández
Ponente: José María Martínez Sánchez

Junio 2018

SEGMENTACIÓN DE OBJETOS EN VÍDEOS NO CONTROLADOS

Ariana Vicario Arroyo
Tutor: Diego Ortego Hernández
Ponente: José María Martínez Sánchez

Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Junio 2018

Trabajo parcialmente financiado por el Ministerio de Economía y Competitividad del Gobierno de España bajo el proyecto TEC2014-53176-R (HAVideo) (2015-2017)

Resumen

En este trabajo, se propone implementar un algoritmo capaz de segmentar objetos de forma automática en secuencias de vídeo grabadas con cámara móvil. Para desarrollarlo se han empleado técnicas del estado del arte que permiten identificar patrones espacio-temporales relevantes y clasificarlos en dos categorías: frente y fondo.

El desarrollo se ha dividido en tres fases. En primer lugar se realiza una segmentación inicial en la que se aplican técnicas de flujo óptico y saliencia para extraer información de movimiento y estimar un modelo de localización. En segundo lugar se lleva a cabo una segmentación media en la que junto al modelo de localización, se estima un modelo de apariencia mediante la utilización de modelos de mezclas de Gaussianas y un Campo Aleatorio Condicional. Finalmente, se realiza una segmentación final donde se estima la calidad del modelo actual y se compara con el mejor modelo obtenido anteriormente, de manera que cuando tiene mayor calidad se actualiza el mejor modelo, mientras que si la calidad es peor se emplea el antiguo modelo para volver a obtener una segmentación.

Para la validación del algoritmo se han realizado experimentos sobre un *dataset* relevante de la literatura estudiado y tanto su parametrización como su rendimiento se comparan con otros algoritmos del estado del arte.

Palabras clave

Saliencia, flujo óptico, superpíxeles, modelo de mezcla de Gaussianas, Campo Aleatorio Condicional .

Abstract

In this work, we propose an algorithm which is able to automatically segment objects from sequences recorded by camera motion. In order to develop it, techniques from the state of art have been used to identify relevant space-temporal patterns and to classify them in two categories: foreground and background.

The development and implementation has been divided into three steps: Firstly, an initial segmentation in which optical flow and saliency techniques have been applied to extract motion information and to estimate a localization model. Secondly, a medium segmentation has been done. With a localization model, an appearance model is estimated by using mixture of gaussian models and a CRF. Finally, a final segmentation, whose quality of the actual model is estimated and it is compared with the best model obtained in previously loops, has been done. Then, the best model is updated when it has a higher quality. Otherwise, if the quality is worst, the old model is used to obtain another segmentation.

In order to validate the algorithm, a set of experiments have been performed to measure the contour accuracy, the region similarity and the temporal stability of the segmented object in each video. As a conclusion, the results of the experiments have been compared with other segmentation algorithms from the research community.

Keywords

Saliency , Optical flow, Gaussian mixture model, Conditional Random Field (CRF)

Agradecimientos

Si retrocediera 5 años en el tiempo probablemente no me habría imaginado escribiendo estos agradecimientos y ahora que me he puesto a redactarlos me doy cuenta de que no es una tarea fácil ya que son muchos los momentos vividos y muchas personas a las que mencionar.

Primero de todo quiero dar las gracias a mi tutor Diego, por toda la ayuda, paciencia y cercanía que ha tenido siempre conmigo. Durante estos meses no solo ha demostrado ser un gran profesor sino que además es una gran persona.

También quiero dar las gracias a todos los amigos que esta carrera me ha dado, a los que empezaron conmigo como a los que he ido conociendo el resto de años. Todos habéis hecho que los momentos tanto dentro como fuera de la universidad sean recuerdos memorables.

A mis compañeros de REE que habéis hecho que las mañanas de trabajo puedan ser también un descanso del estudio.

Y por supuesto quiero agradecérselo a mi familia. A mis padres por quererme, guiarme y enseñarme desde pequeña que nunca hay que rendirse a pesar de los obstáculos que te encuentres. Y por último a mis hermanas Ángela y Ester, en especial a Ester, sin la cual ahora mismo yo no estaría escribiendo estás lineas. Aprovecho este momento para decirte que siempre has sido y serás un gran ejemplo a seguir.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este apartado introductorio se explica la motivación y los objetivos principales que se plantean en este Trabajo Fin de Grado.

1.1. Motivación

La visión artificial (en inglés, *computer vision*) tiene como objetivo programar una máquina para que ésta sea capaz de percibir el mundo que le rodea igual o mejor que un ser humano. Los sistemas basados en técnicas de visión artificial permiten adquirir, de manera automática, imágenes para poder procesarlas, analizarlas y comprender los datos que tienen dichas imágenes para actuar en consecuencia.

En la actualidad, la visión artificial está cobrando una gran importancia debido a una gran evolución tecnológica. Esto se observa en la infinidad de aplicaciones prácticas que existen en campos como la agricultura, biometría, realidad aumentada, restauración de imágenes, análisis de imágenes médicas, teledetección, robótica, seguridad, etc. Por lo tanto, conseguir un rendimiento significativo implica desarrollar algoritmos que mejoren el trabajo realizado previamente.

La segmentación de objetos en vídeos es una tarea de gran interés en visión artificial, ya que tiene una gran utilidad para nutrir etapas posteriores de seguimiento de objetos o detección de actividades, además de una aplicabilidad para tareas de edición de vídeo y otras muchas tareas. Existen diferentes técnicas para segmentar un vídeo en regiones que estén claramente diferenciadas [10] (ver Figura 1.1). Partiendo del nivel de segmentación más sencillo al más complejo las técnicas se pueden clasificar en: segmentación con superpíxeles [5][3], segmentación frente-fondo [2], segmentación de instancias (objetos-fondo pero asignando a cada objeto una etiqueta de clase a la vez que se distingue entre instancias distintas de una misma clase) [9], segmen-

tación semántica (se etiquetan todas las clases de objetos del frente y del fondo sin distinguir instancias de una misma clase) [27], segmentación panóptica (extensión de segmentación semántica pero distinguiendo entre instancias de una misma clase) [13] o segmentación semántica de acciones [26].

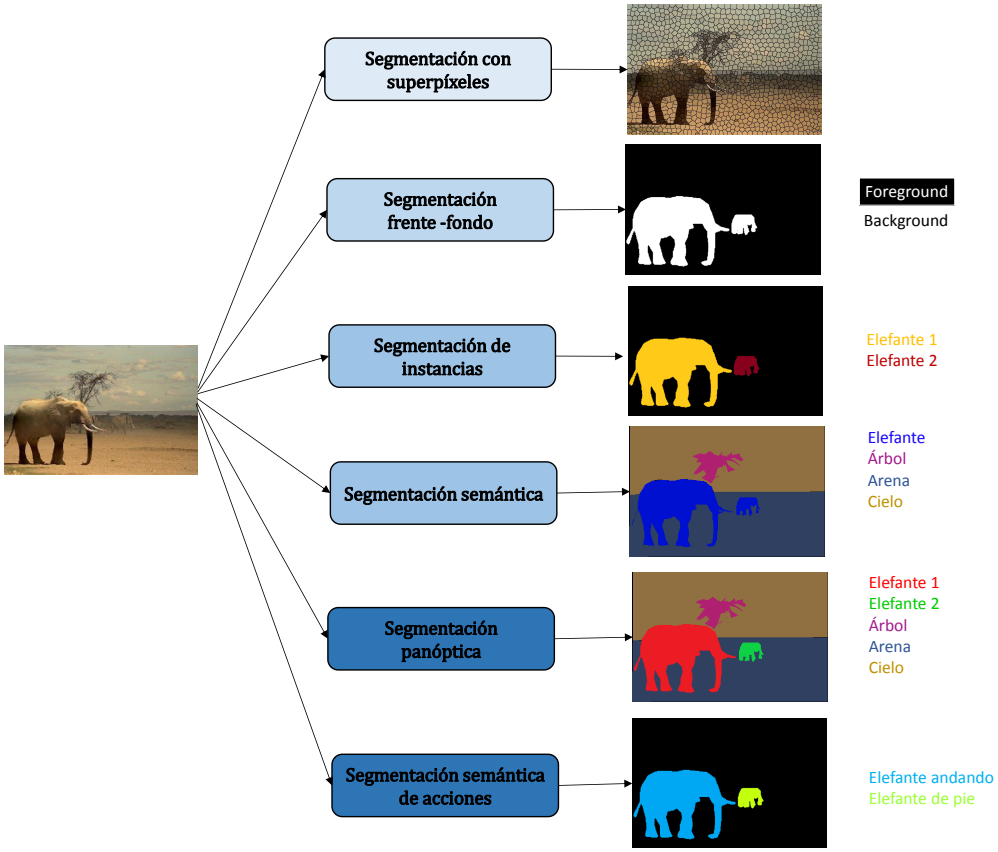


Figura 1.1: Tareas de segmentación con conocimiento semántico.

Entre estas tareas de segmentación, la segmentación frente-fondo tiene como objetivo detectar objetos de interés frente al fondo de la escena. En imágenes, el primer plano o frente se define como un objeto saliente o aquellas apariencias con un elevado *objectness* [1] mientras que en un vídeo suele corresponder con los objetos en movimiento [25] .

En este Trabajo Fin de Grado se va a trabajar en la segmentación frente-fondo en secuencias de vídeo con cámara móvil donde la dificultad de la tarea reside en abordar el movimiento de la cámara, así como las deformaciones que pueden sufrir algunos objetos. Por ello, conseguir algoritmos capaces de realizar esta tarea de forma automática es un gran reto.

1.2. Objetivos

El objetivo principal de este TFG es segmentar objetos en secuencias de vídeos con cámara móvil de manera automática mediante el uso de técnicas del estado del arte. El objetivo principal se ha dividido en los siguientes sub-objetivos:

1. Estudio del estado del arte en segmentación frente-fondo para conocer las posibilidades que ofrece desarrollar un algoritmo de esta modalidad.
2. Diseño e implementación de un algoritmo para la segmentación frente-fondo en secuencias con cámaras móviles.
3. Evaluación del rendimiento del algoritmo en *datasets* de referencia del estado del arte.

1.3. Organización de la memoria

La memoria se organiza por capítulos y éstos contienen diferentes sub-secciones:

- **Capítulo 1**: Introducción: Motivación del trabajo y objetivos.
- **Capítulo 2**: Estado del estado: Estado del arte sobre la segmentación de objetos en secuencias de vídeo.
- **Capítulo 3**: Diseño y desarrollo: Implementación del algoritmo con diferentes técnicas explicadas en el estado del arte.
- **Capítulo 4**: Trabajo experimental: Información de los *datasets* y las métricas empleadas para su evaluación. Comparativa de los resultados obtenidos con otros algoritmos no supervisados.
- **Capítulo 5**: Conclusiones y trabajo futuro: Reflexiones personales y mejoras que se podrían incluir en este tipo de sistemas.
- **Bibliografía**: Bibliografía que incluye trabajos relacionados relevantes para la realización de este TFG.

Capítulo 2

Estado del arte

En este capítulo se hace un análisis del estado del arte y se explican los diferentes tipos de algortimos existentes para la segmentación frente-fondo.

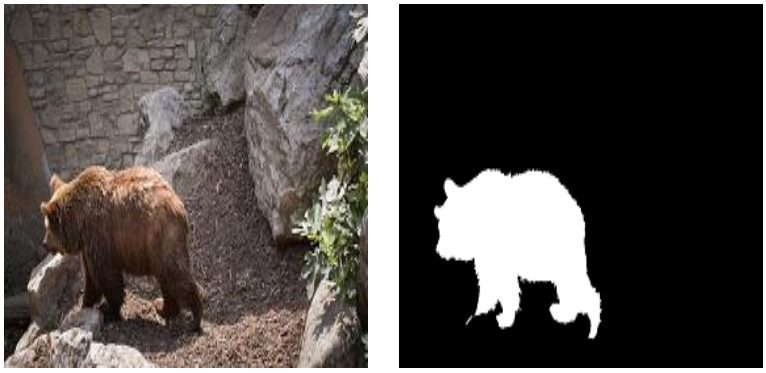
2.1. Introducción

La segmentación frente-fondo es una tarea específica de entre todas las tareas de segmentación (ver Capítulo 1) que consiste en segmentar los objetos de interés, es decir, segregar objetos de primer plano o *foreground* del fondo de la escena o *background*. Los algortimos desarrollados en la literatura pueden clasificarse en dos grandes grupos en función del dominio de aplicación: cámara estática o cámara móvil.

2.2. Segmentación frente-fondo con cámara estática

De entre los segmentadores más conocidos destacan los de modelado de fondo. Se encargan de comparar la imagen que se está analizando con un modelo de fondo para obtener una máscara de frente. El más conocido es el de tipo Sustracción de Fondo o *Background Subtraction* (BGS), que genera un modelo de fondo que después se empleará para obtener el primer plano de la imagen original, es decir, compara una imagencon el modelo de fondo de la escena para obtener la segmentación del objeto (ver Figura 2.1).

De entre las fases principales que debe tener un algoritmo de estas características destacan: Inicialización de fondo, modelado de fondo, actualización del modelo y detección de frente. Un aspecto importante de BGS es la selección de las características empleadas para la segmentación frente-fondo [2]. Estas características pueden ser información de color, texturas, movimiento u otras representaciones más complejas,



(a) (b)

Figura 2.1: Ejemplo de una imagen con su máscara frente.

representando una combinación de múltiples características la opción más robusta para diseñar un algoritmo de BGS. Por otro lado, es importante también considerar el tipo de retos que se dan en los algoritmos de BGS: cambios de iluminación, fondos dinámicos, camuflajes, *jitter* de la cámara, oclusiones y sombras. Típicamente los algoritmos desarrollan estrategias para conseguir ser robustos a estos problemas, así por ejemplo para lidiar con fondos dinámicos se utilizan modelos con múltiples representaciones o para lidiar con cambios de iluminación se diseñan estrategias de actualización del modelo de fondo para conseguir adaptarse a las variaciones espacio-temporales de la escena. En [2] se analizan en detalle los algoritmos de BGS así como sus retos y configuraciones más comunes.

2.3. Segmentación frente-fondo con cámara móvil

Cuando se trabaja en entornos con cámara móvil, el modelado del fondo realizado en entornos con cámara fija se hace inviable, lo cual lleva a descartar las aproximaciones de BGS y obliga a orientar las técnicas en una dirección diferente. De entre los problemas que introducen los entornos con cámara móvil destacan: el emborronamiento de imágenes debido al movimineto, oclusiones y deformaciones de los objetos.

Los algoritmos de segmentación frente-fondo con cámara en movimiento buscan detectar patrones espacio-temporales relevantes asociados a objetos. La manera de definir dicha relevancia permite organizar los algoritmos en 3 categorías [11]: no supervisados , semi-supervisados y supervisados.

Los algoritmos no supervisados [4] son técnicas automáticas que no requieren inicialización manual. Estos algoritmos definen la relevancia del patrón espacio-temporal a segmentar mediante su movimiento [18] o saliencia visual [24] (ver Figura 2.2). Res-

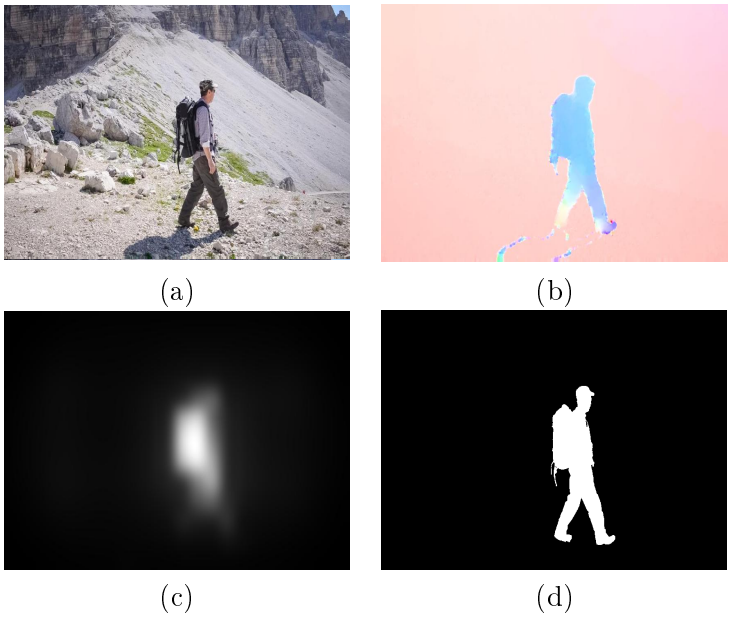


Figura 2.2: Ejemplo del algoritmo empleado. a) Imagen de entrada, b) su correspondiente flujo óptico, c) saliencia en movimiento y d) su correspondiente máscara

pecto a los algortimos semi-supervisados [11], requieren una inicialización manual en el primer *frame* del objeto u objetos de interés, de manera que los patrones espacio-temporales asociados a dichos objetos son los de interés en el resto del vídeo. Este tipo de algoritmos, tienen un marco de funcionamiento muy parecido al de los algoritmos de seguimiento de objetos [16]. Finalmente, los algoritmos supervisados [15], requieren de etiquetado manual en todo momento, funcionando de forma iterativa durante todo el proceso de segmentación. El último tipo de algoritmos consiguen producir una segmentación fina, siendo muy utilizados en tareas de post-producción.

Capítulo 3

Algoritmo propuesto

En este capítulo se explica cómo a partir de varias herramientas del estado del arte se ha desarrollado un algoritmo no supervisado que se encarga de segmentar objetos de secuencias de vídeo con cámara móvil.

3.1. Esquema general

En esta sección se describe un esquema general reflejado en el diagrama de bloques de la Figura 3.1. La idea base de este algoritmo no supervisado consiste en obtener una segmentación lo más precisa posible identificando patrones espacio-temporales y clasificarlos como frente y fondo. En cada uno de los bloques se detallan los problemas o soluciones propuestas así como las referencias a las técnicas del estado del arte empleadas. El algoritmo implementado tiene como finalidad segmentar un objeto mediante un etiquetado frente-fondo que se abordará de manera no supervisada. A grandes rasgos, el algoritmo se divide en tres bloques o segmentaciones principales que se detallan a continuación:

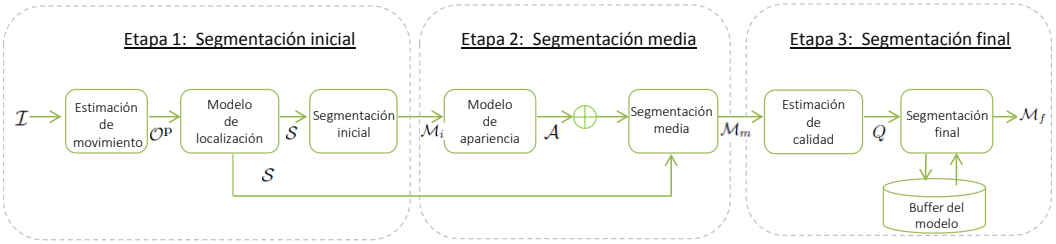


Figura 3.1: Diagrama de bloques del sistema implementado

1. **Segmentación inicial:** partiendo de una secuencia de vídeo de entrada, se procede a calcular un módulo de localización que atribuye a cada pixel la pro-

babilidad de pertenecer a un modelo de frente. Para obtener este modelo de localización, se realiza una estimación de movimiento en la que se emplea como técnica el flujo óptico [23] y partiendo de esta se utiliza un marco de saliencia visual para detectar las zonas en movimiento [8]. Empleando ambas técnicas, se consigue una segmentación inicial.

2. **Segmentación media:** se basa en calcular la apariencia del objeto, el cual atribuye la probabilidad de que cada pixel pertenezca a un modelo de frente, pero esta vez empleando un modelo de mezcla de Gaussianas. Este modelo de apariencia junto con el modelo de localización permitirá obtener las energías finales para cada pixel. A continuación para obtener una segmentación media, se aplica la técnica *MeanField* que se encarga de obtener el valor ideal de esta energía utilizando un campo aleatorio condicional (CRF).

3. **Segmentación final:** consiste en obtener una segmentación final, para ello se basa en estimar la calidad del modelo actual y se compara con el mejor modelo obtenido anteriormente, de manera que cuando tiene mayor calidad se actualza el mejor modelo, mientras que si la calidad es peor se emplea el antiguo modelo para volver a obtener una segmentación.

3.2. Descripción del algoritmo

3.2.1. Segmentación inicial: Modelo de localización

Debido al carácter no supervisado con el que se aborda la tarea de segmentación frente-fondo, el primer paso para segmentar los objetos de interés en una imagen $\mathcal{I}$ es localizarlos de manera automática. En particular, los objetos de frente suelen tener un movimiento que contrasta en magnitud o dirección con el movimiento de la cámara. Por tanto, se ha decidido utilizar información de flujo óptico u *optical flow* [23], pues proporciona para cada pixel p su vector de movimiento $\mathcal{O}^p$, de manera que el mapa 2-D de movimiento puede definirse como $\mathcal{O} = \{\mathcal{O}^p\}_{\forall p \in \mathcal{I}}$. En la Figura 3.2 se presentan tres ejemplos de mapas de flujo óptico $\mathcal{O}$, donde puede observarse como la magnitud y dirección del movimiento para los objetos y la cámara tienen cierto contraste. Es importante destacar que las imágenes de flujo óptico se presentan en su formato de color (la intensidad del color está relacionada con la magnitud del flujo óptico y el color con la dirección de dicho movimiento), transformación que es habitual en la literatura [11].

Para la estimación de movimiento se ha empleado el flujo óptico [23] y a partir de éste, se ha decidido utilizar un marco de saliencia visual para detectar las zonas

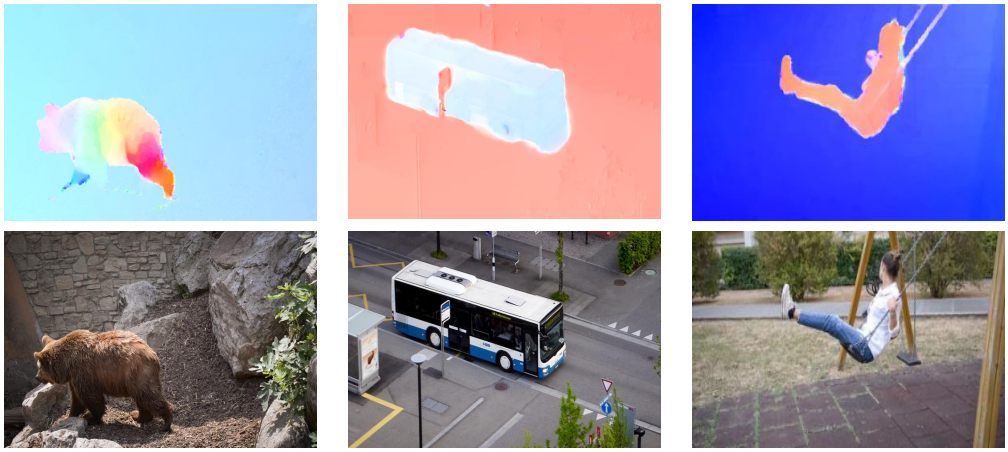


Figura 3.2: Ejemplos del mapa de flujo óptico $\mathcal{O}$ en diferentes secuencias de vídeo. En la primera fila se incluye el flujo óptico $\mathcal{O}$ y en la segunda la imagen asociada a dicho flujo.

salientes de la imagen en cuanto a movimiento se refiere. La saliencia o *saliency* es una representación de la intensidad relativa de la respuesta a estímulos en el espacio visual, es decir, las zonas de las imágenes más relevantes o que más destacan y en las que el ojo humano centra más su atención. Dicha saliencia suele calcularse sobre espacios de color y se basa en encontrar zonas que localmente contrastan con su entorno. Esta idea puede ser aplicada también a otros espacios de características como el movimiento para encontrar dichas zonas contrastadas. En concreto, se ha seleccionado el algoritmo GVBS (*Graph-Based Visual Saliency*) [8], para obtener un mapa de saliencia $\mathcal{S}$ a partir del flujo óptico:

$$\mathcal{S} = f\left(\mathcal{O}\right), \tag{3.1}$$

donde $f\left(\cdot\right)$ es la función para calcular la saliencia propuesta en [8]. Es importante destacar que el algoritmo puede aplicarse directamente gracias a que se tiene el flujo óptico en formato de color. Por último, para tener coherencia espacial en la información de saliencia y asemejar la saliencia a un mapa de probabilidades se realiza una normalización respecto al valor máximo y una agregación espacial para generar el mapa de probabilidades de saliencia $\mathcal{L}$ a nivel de superpixel [7]. Para ello se ha asignado a cada píxel de cada superpixel, la saliencia media de todos los píxeles del superpixel. El efecto de los parámetros de esta etapa se analiza en las subsecciones 4.2.1 y 4.2.4, respectivamente, incluyendo experimentos para estudiar la manera de agregar la saliencia a nivel de superpixel y la parametrización de la segmentación en superpíxeles (dependiente de un parámetro α que controla el grado de detalle en la

segmentación).

En esta etapa se obtienen dos salidas: una máscara de segmentación inicial $\mathcal{M}_i$ y un mapa de probabilidades de saliencia $\mathcal{S}$. Este mapa de probabilidades proporciona la probabilidad de pertenencia de que cada pixel p perteneza a $\mathcal{M}_i$. El cálculo de $\mathcal{M}_i$ consiste en umbralizar $\mathcal{S}$ usando un valor β (este parámetro esta sujeto a experimento (ver Sección 4.2.2)).Es decir, se obtiene una máscara inicial de frente $\mathcal{M}_{iFG}$ cuando se tome $\mathcal{S} > \beta$ y una máscara inicial de fondo $\mathcal{M}_{iBG}$ cuando $\mathcal{S} < \beta$. Estas máscaras definen una segmentación inicial o grosera del objeto que está en movimiento (ver Figura 3.3).

3.2.2. Segmentación media: localización y apariencia

Como se evaluó en la (Sección 3.2.1), el modelo de localización permite obtener una segmentación inicial en la cual se averigua la localización del objeto en movimiento. Pero con este modelo no es suficiente, es necesario calcular un modelo de apariencia con el cual se modele la apariencia del frente y del fondo y junto con el de localización se obtengan potencias individuales por píxel para obtener la segmentación media.

El modelo de apariencia no estima a nivel de pixel, sino estima a nivel de regiones. Una región $|\mathcal{R}_i|$ es un grupo de píxeles conectados espacialmente que representan una parte del espacio que viene definida por $\mathcal{R}_i = \{\mathcal{R}^{\mathbf{p}}\}_{\forall \mathbf{p} \in \mathcal{I}}$. Estas $|\mathcal{R}_i|$ se encargan de modelar la información que define cual es el color que más predomina de los objetos en movimiento empleando dos modelos de mezclas de Gaussianas (GMM). Los GMM corrigen las malas correspondencias que se han producido en el modelo de localización dependiendo de los niveles de color RGB generando un modelo de distribución sobre los datos de una imagen en el espacio RGB. Estos GMM proponen como solución modelar la intensidad de los píxeles con una mezcla de λ distribuciones Gaussianas (donde λ generalmente es un número de 3 a 7 Gaussianas) que a su vez están definidos por tres componentes: una media μ, una matriz de covarianzas σ y unos pesos asociados ρ.

Por tanto los modelos de apariencia $\mathcal{A}$ crean dos mapas de probabilidades $\mathcal{A}_{BG}$ y $\mathcal{A}_{FG}$ en el que atribuyen la probabilidad de que una región $|\mathcal{R}_i|$ pertenezca a una etiqueta $e_{\mathcal{R}i}$ de fondo o frente respectivamente mediante:

$$\mathcal{A}(\mathcal{R}_i | e_{R_i} = 0) = \mathcal{A}_{BG}(p) \tag{3.2}$$

$$\mathcal{A}(\mathcal{R}_i | e_{R_i} = 1) = \mathcal{A}_{FG}(p) \tag{3.3}$$

Una vez que se tiene disponible el modelo de localización (ver Sección 3.2.1) y de apariencia (ver Sección 3.2.2) se implementa un sistema de potencias individuales

que compone ambos modelos dando lugar a una energía unitaria de cada región de la imagen. Para calcular esta energía unitaria u_{Ri} es necesario aplicar:

$$u_{Ri}(e_{\mathcal{R}i}) = \text{- } log\ \mathcal{A}(\mathcal{R}_i \mid e_{\mathcal{R}_i}) \text{ - } log\ \mathcal{L}(e_{\mathcal{R}_i}), \tag{3.4}$$

Esta energía u_{Ri} indica la probabilidad de que a una región $\mathcal{R}_i$ se le asigne una etiqueta de frente o fondo de acuerdo a un modelo de apariencia $\mathcal{A}$ y a un modelo de localización $\mathcal{L}$. Para obtener la segmentación media $\mathcal{M}_m$ se propone segmentar mediante minimización de energías, para ello se calcula una energía total E que viene definida como:

$$E = \sum_{\mathcal{R}_i} u_{R_i}(e_{\mathcal{R}_i}) + \sum_{\mathcal{R}_i} v_{\mathcal{R}_i,R_j}(e_{\mathcal{R}_i}, e_{\mathcal{R}_j}), \tag{3.5}$$

donde p es cada pixel de $\mathcal{I}$ y las energías individuales y conjuntas son u_{Ri} y $v_{Ri,Rj}$ respectivamente. Para encontrar la segmentación ideal de esta función de energía total se aplica un CRF (*Conditional Random Field*) [20] que se encarga de hacer el etiquetado de las muestras incorporando información de espacio-tiempo entre los distintos *frames*. Una vez que se ha aplicado este CRF se obtendrá una segmentación media $\mathcal{M}_m$, que es una nueva máscara con unos resultados más refinados que los que se obtienen en 3.2.1

3.2.3. Segmentación final: Refinamiento: selección automática de modelo

Una vez que se ha obtenido la segmentación media $\mathcal{M}_m$, se pretende obtener una segmentación final $\mathcal{M}_f$ en la que se va hacer un refinamiento de la segmentación actualizando los modelos en el tiempo. La finalidad de esta actualización es segmentar con el mejor modelo que se haya obtenido.Para ello, es necesario la obtención de la mejor calidad del modelo $\mathcal{Q}_b$, que se estima realizando un ajuste de las regiones $|\mathcal{R}_i|$, cuanto mayor sea el ajuste de las regiones, la calidad del modelo en ese instante temporal $\mathcal{Q}_t$será mejor. El problema surge al estimarlo a nivel de región, ya que existen un gran número de regiones y por ende un gran número de calidades q_i. Por consiguiente se propone hacer un ajuste de dichas regiones a un sólo valor $\mathcal{Q}$, como la media ponderada de los píxeles de frente $\mathcal{M}_m^p$, es decir, la calidad del modelo $\mathcal{Q}$ se calcula como:

$$\mathcal{Q} = \sum_{\forall p \in R_i} \frac{\mathcal{M}_m^p}{|\mathcal{R}_i|}, \tag{3.6}$$

Por otro lado, el modelo de segmentación$\mathcal{X}$ se define como:

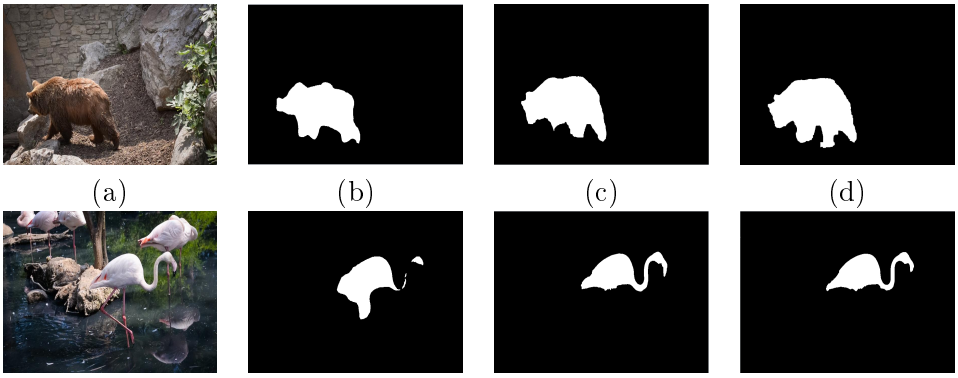
$$\mathcal{X} = \begin{cases} \mathcal{X}_t & si \quad \mathcal{Q}_t \geq \mathcal{Q}_b \\ \mathcal{X}_b & si \quad \mathcal{Q}_t < \mathcal{Q}_b \end{cases}, \quad (3,7)$$

siendo $\mathcal{X}_t$ el modelo con el que se está segmentando en ese instante y $\mathcal{X}_b$ el mejor modelo de segmentación obtenido en instantes anteriores.

En cada instante t se extraen cuatro cálculos: $\mathcal{X}_{FGt}$, $\mathcal{X}_{BGt}$, $\mathcal{S}_t$ y $\mathcal{Q}_t$ que son el modelo de frente, modelo de fondo, saliencia y calidad de ese instante temporal con los que se esta realizando la segmentación media. Además, se va a trabajar con otros cuatro cálculos más, obtenidos de secuencias anteriores: $\mathcal{X}_{FGb}$, $\mathcal{X}_{BGb}$, $\mathcal{S}_b$y $\mathcal{Q}_b$ que corresponden al mejor modelo de frente, mejor modelo de fondo, mejor saliencia y mejor calidad respectivamente. Por ello, para conseguir un buen refinamiento se va a realizar sólo una comparación de las calidades. Si la calidad $\mathcal{Q}_t \geq Q_b$, indica que los modelos con los se está obteniendo $\mathcal{M}_m$ en ese instante son los correctos y por ello $\mathcal{M}_f = \mathcal{M}_m$. En este caso se procede actualizar los mejores modelos, calidad y saliencia, es decir $\mathcal{X}_{FGb} = \mathcal{X}_{FGt}$, $\mathcal{X}_{BGb} = \mathcal{X}_{BGt}$, $\mathcal{S}_b = \mathcal{S}_t$ y $\mathcal{Q}_b = \mathcal{Q}_t$.

Por el contrario, si $\mathcal{Q}_t < Q_b$, indica que los modelos con los que se está obteniendo $\mathcal{M}_m$ no son los más óptimos y para obtener $\mathcal{M}_f$ es necesario realizar de nuevo la segmentación es este instante pero con los mejores modelos que se han estimado $\mathcal{X}_{FGb}$, $\mathcal{X}_{BGb}$, $\mathcal{S}_b$ y $\mathcal{Q}_b$. Durante la actualización del modelo es necesario emplear un parámetro φ. Este parámetro también se va a someter a experimento (ver Sección 4.2.5). Este parámetro tiene la finalidad de actualizar a su vez los mejores modelos de calidad $\mathcal{Q}_b$ que se están almacenando, es decir, si $\mathcal{Q}_b$ no se actualiza en varios instantes seguidos, llegará un momento en que esta calidad será obsoleta. Por este motivo se multiplica φ por $\mathcal{Q}_b$, haciendo que en cada instante disminuya la calidad $\mathcal{Q}_b$ para que se vaya actualizando más frecuentemente.

En la Figura 3.3 se muestran dos ejemplos de las diferentes segmentaciones que se han realizado: inicial, media y final.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figura 3.3: Representación de los tipos de segmentación: (b) y (f) corresponden a la segmentación inicial, (c) y (g) corresponden a la segmentación media, (d) y (h) corresponden a la segmentación final.

Capítulo 4

Trabajo experimental

En este capítulo se evalúa el sistema desarrollado con el fin de verificar su funcionalidad. En primer lugar. se explica el *dataset* empleado junto con las métricas que se han utilizado para su evaluación. A continuación. se plantean una serie de experimentos para decidir la parametrización del algoritmo y finalmente se compara el rendimiento obtenido frente a otros algoritmos del estado del arte.

4.1. Marco de evaluación

4.1.1. *Dataset*

Para realizar la evaluación del algoritmo se ha empleado el *dataset* DAVIS (*Densely Annotated VIdeo Segmentation*) 2016 [21][19]. Fue creado con el fin de tener un conjunto de datos equilibrado y completo para usarse en tareas de segmentación de objetos de vídeo. En todas las secuencias aparece un objeto a segmentar que tendrá un movimiento característico que permite diferenciarlo del resto de la escena. En cuanto a los datos. el *dataset* DAVIS consta de 50 secuencias de vídeo con objetos de 4 clases principales: personas. animales. objetos y vehículos. Dicho *dataset* es lo suficientemente grande para asegurar una diversidad del contenido y proporcionar un conjunto de desafíos uniforme. Todas las secuencias tienen una corta duración (aproximadamente de 2 a 4 segundos). es decir. en media unos 70 *frames*. En ellas se abarcan varios retos puesto que aparecen objetos más pequeños. estructuras más finas. cambios de apariencia. movimiento más rápido y oclusiones. Con respecto a la resolución. las 50 secuencias están capturados a 24 fps estando disponibles las resoluciones 1080p. 720p y 480p. En este trabajo se han empleado las secuencias a resolución de 480p para no incrementar el tiempo de ejecución.

4.1.2. Métricas

Para la evaluación de los resultados se han utilizado tres métricas distintas: región de similitud $\mathcal{J}$. establidad temporal $\mathcal{T}$ y precisión del contorno $\mathcal{F}$. Algunas de ellas se encuentran explicadas en [19]. A continuación se definen cada una de ellas:

- $\boldsymbol{\mathcal{J}}$ mide la región de similitud. es decir como de bien coinciden los píxeles de dos máscaras. Para ello se estima el número de píxeles que están mal etiquetados. Esta estimación se realiza mediante el índice *Jaccard*. que se define como la intersección entre la unión de la segmentación obtenida y la máscara de *Ground-truth*.

  $$\mathcal{J} = \frac{|M \cap G|}{|M \cup G|},$$

  siendo M la segmentación obtenida. es decir la máscara y G la máscara de *Ground-Truth* (máscara de segmentación con precisión de pixel de la imagen).

- $\boldsymbol{\mathcal{F}}$: indica como de preciso es el contorno. es decir. evalúa cómo de exactos son los límites a través de la coincidencia entre los contornos de la máscara segmentada y la de ground-truth.

- $\boldsymbol{\mathcal{T}}$: indica la estabilidad temporal. El objetivo es conseguir distinguir un movimiento aceptable de los objetos y de las fluctuaciones no deseadas. Por ello se estima la deformación para transformar la máscara de un *frame* a otro. Si la transformación es suave y precisa el resultado es estable. En muchas secuencias del *dataset* aparecen oclusiones lo que hace que los resultados sean menos significativos.

4.2. Parametrización del algoritmo

En este apartado se explica cada uno de los experimentos realizados para los cuales se obtiene una parametrización del algoritmo implementado. Se han realizado 6 experimentos. en los cuales se han modificando parámetros o bien comprobado que métodos son los más idóneos para obtener una segmentación mejor. A medida que se realizaba un experimento. el valor con el que se obtenía el mejor resultado se queda fijado como mejor parámetro para el resto de experimentos. En cuanto a las secuencias utilizadas para la parametrización. se han escogido 9 secuencias del *dataset* DAVIS [21]. eligiendo secuencias donde haya una variedad. es decir. que salgan 3 personas. 3 animales y 3 vehículos. Para los primeros cuatro experimentos se han modificado los distintos parámetros. mientras que para el último experimento se ha evaluado

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Opción 1	0.5329	0.1482	0.3636	0.1261	0.7542
Opción 2	**0.5382**	**0.1594**	**0.4094**	**0.1370**	**0.7808**

Tabla 4.1: Elección de la saliencia a nivel se superpixel. Ya sea. opción 1: del máximo entre la saliencia a nivel de superpixel o saliencia u opción 2: saliencia a nivel se superpixel. La mejor opción ha sido la 2.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
	Mean↑	Std↓	Mean↑	Std↓	Mean↓
β =0.1	0.5184	0.1570	0.3741	0.1336	0.9530
β =0.2	0.5447	0.1433	0.3998	0.1276	0..8071
β =0.3	0.5364	0.1524	0.4073	0.1357	0.7915
β =0.4	**0.5382**	**0.1594**	**0.4094**	**0.1370**	**0.7808**
β =0.5	0.5235	0.1629	0.3948	0.1352	0.7969
β =0.6	0.4793	0.1719	0.3616	0.1405	0.8075

Tabla 4.2: Experimento 2: barrido del parámetro β o umbral de la saliencia. Se observa que la mejor configuración se obtiene para β =0.4.

todo el *dataset* salvo las 9 secuencias anteriores utilizando la mejor parametrización encontrada.

Los parámetros que se van a modificar para observar como varía la segmentación son: la saliencia a nivel de superpixel (Sección 3.2.1). el parámetro β o umbral de la saliencia (ver Sección 3.2.1). el parámetro λ o número de Gaussianas (ver Sección 3.2.2). α o parámetro de *merging* (ver Sección 3.2.1) y el parámetro φ o tasa de actualización (ver Sección 3.2.3). Como punto de partida se fijan los parámetros 0.4 para β. 3 para λ. 0.07 para el parámetro α y para φ no se seleccionan tasa de actualización.

4.2.1. Experimento 1: Elección de la saliencia a nivel de superpixel

El primer experimento ha consistido en evaluar la manera de obtener mejores resultados de la sección 3.2.1. Para calcular la saliencia a nivel de superpixel surgen dos posibles formas de obtención. La primera de ellas consiste en asignar la saliencia a nivel de superpixel como el máximo entre la saliencia o la saliencia a nivel de pixel. La segunda. por el contrario. consiste solamente en coger la saliencia a nivel de superpixel. Finalmente al observar los resultado (ver Tabla 4.1). por media. se observa que es mejor escoger la segunda opción ya que para la precisión del contorno se obtienen unos resultados mejores. Para los sucesivos experimentos se fijará este parámetro como el valor óptimo.

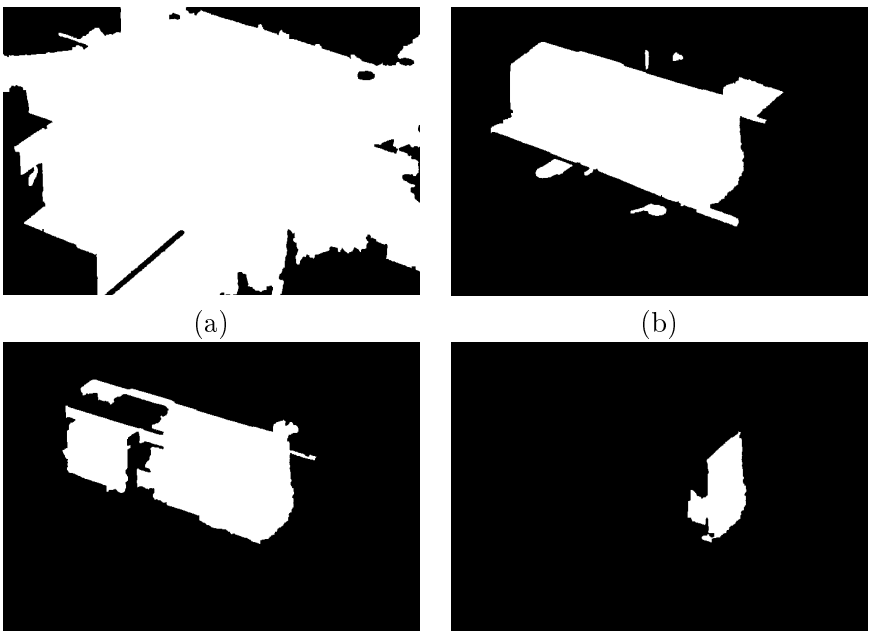


Figura 4.1: Resultados obtenidos para las siguientes secuencias modificando el parámetro β: (a) $\beta = 0.1$ (b)$\beta = 0.3$ (c)$\beta = 0.5$ (d)$\beta = 0.7$. A la vista de los resultados se observa que para esta secuencia el mejor parámetro con el que mejor se segmenta es para $\beta = 0.3$

4.2.2. Experimento 2: Selección del umbral de la saliencia β

El segundo experimento ha consistido en evaluar el umbral de la saliencia β (ver Sección 3.2.2). Este parámetro β permite crear la primera estimación para los modelos de frente y de fondo. es decir. si los píxeles que seleccionamos son mayores a ese umbral. estos pertenecerán al modelo de fondo. en caso contrario pertenecerán al de frente. En la Figura 4.1 se muestra una representación de una secuencia en la que ha variado β. Este parámetro cobra una gran importancia a la hora de realizar la primera segmentación ya que con el se definen las máscaras de frente y de fondo. La franja de valores experimentados ha sido del 0.1 al 0.6 en pasos de 0.1. Todos los resultados se muestran en las siguientes tablas en la sección de anexos (ver Tablas 24. 25. 26. 27. 28 y 29). A la vista de los resultados. se observa que el β que mejor parametriza las nueve secuencias es 0.4. ya que es el umbral que mejor clasifica los píxeles como frente y fondo. Para los casos en los que β es un valor bajo clasifica un gran número de pixeles de fondo como píxeles de frente. De igual modo. en los que β es una valor alto. se clasifican un gran número de pixeles de frente como píxeles de fondo. Para los sucesivos experimento se fija un β con valor 0.4.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
	Mean↑	Std↓	Mean↑	Std↓	Mean↓
λ= 3	0.5384	0.1592	0.4092	0.1366	0.7818
λ= 5	0.5688	0.1590	0.4198	0..1432	0.8487
λ= 7	0.5726	0..1598	0.4202	0.1392	0.8574

Tabla 4.3: Experimento 3: barrido del parámetro λ

4.2.3. Experimento 3: selección del número de Gaussianas λ

El tercer experimento ha consistido en realizar pruebas para los casos en los que modelamos con 3. 5 y 7 Gaussianas.

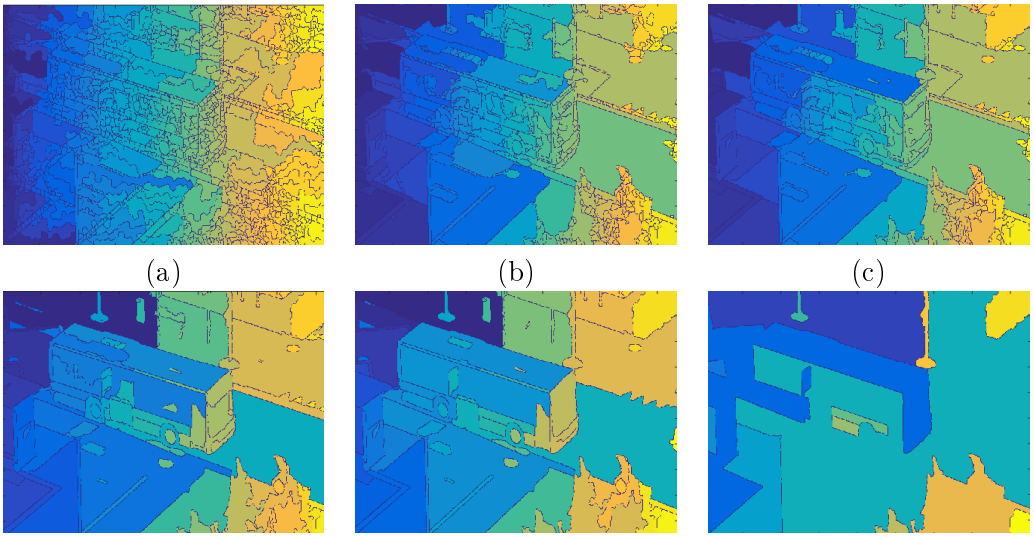
En la sección 3.2.2. se analizó la importancia de escoger un número de Gaussianas adecuado para configurar cada modelo. Este parámetro λ es un factor clave para conseguir el correcto funcionamiento del algoritmo ya que si se modela con un bajo número de Gaussianas puede ocasionar pérdida de información y por ende la errónea segmentación del objeto. Del mismo modo. modelar con muchas Gaussianas va a requerir un coste computacional elevado y por tanto. el tiempo de ejecución también será elevado. A la vista de los resultados se observa que a medida que el número de Gaussianas aumenta los resultados que se obtienen son mejores. pero su tiempo de ejecución también. Por ello se concluye que la elección de 5 Gaussianas es la más óptima ya que los resultados (ver tabla 4.3) y el tiempo de ejecución son bastante asequibles para conseguir un equilibrio entre ambas opciones. Los resultados del detalle de las secuencias se muestran en las siguientes tablas de los anexos (30. 31 y 32).

4.2.4. Experimento 4: Selección del parámetro de *merging* α

El cuarto experimento ha consistido en modificar el α. este parámetro indica cómo se van a fusionar las regiones de los superpíxeles que se usarán para realizar la segmentación. Como se muestra en (Figura 4.2).

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
	Mean↑	Std↓	Mean↑	Std↓	Mean↓
$\alpha = 0{,}01$	0.5569	0.1453	0.3600	0.1279	0.7721
$\alpha = 0{,}03$	0.5679	0.1494	0.3890	0.1331	0.8184
$\alpha = 0{,}05$	0.5715	0.1567	0.4136	0.1398	0.8183
$\alpha = 0{,}07$	0.57260	0.1598	0.4202	0.1392	0.8574
$\alpha = 0{,}09$	**0.5776**	**0.1637**	**0.4326**	**0.1434**	**0.8641**
$\alpha = 0{,}11$	0.5762	0.1672	0.4293	0.1449	0.884

Tabla 4.4: Experimento 4: barrido del parámetro α



(a) (b) (c)

(d) (e) (f)

Figura 4.2: Creación de regiones en función del parámetro α. Los valores son los correspondientes a: a = 0.01. b = 0.05. c = 0.09. d = 0.2. e = 0.3. f = 0.6. Se observa que cuanto mayor sea este parámetro el número de regiones es menor y por ende son más grandes.

Cuanto más bajo sea este parámetro α. las regiones se formarán con un menor número de superpíxeles y viceversa. Las pruebas se han realizado tomando el parámetro desde el valor 0.01 hasta el 0.11 en pasos de 0.02. Para observar la evaluación de cada una de las secuencias más en detalle ver las tablas de los anexos (33. 34. 35. 36. 37 y 38). El mejor valor del parámetro α es el que se obtiene para 0.09 (ver Tabla 4.4). ya que se puede apreciar como las regiones que se forman tienen un número adecuado de superpíxeles siendo las regiones lo suficientemente robustas para poder realizar una segmentación más óptima. Por ello se concluye que el mejor α es 0.09 y se fijará como mejor opción para los siguientes experimentos. A partir del quinto experimento se han actualizado los modelos en el tiempo. en los cuatro anteriores no.

4.2.5. Experimento 5: Frecuencia de actualización de modelos φ

En el quinto experimento se ha evaluado el parámetro φ (ver sección 3.2.3). Para ello se han hecho pruebas para comprobar si es mejor actualizar modelos en el tiempo o no. Y en caso de ser mejor con que frecuencia se consiguen mejores resultados. Este parámetro tiene la finalidad de actualizar a su vez los mejores modelos de calidad que se están almacenando. es decir. si la calidad del modelo con el que se está comparando es siempre la misma. llegará un momento en que esta va a ser obsoleta. Por este motivo esta tasa será un β que multiplicará a la calidad del modelo. haciendo que en

cada modelo se disminuya el valor de esta calidad para que se vaya actualizando más frecuentemente. Por ello se concluye que mediante la actualización de los modelos se consigue una mejora notable en los resultados. Además de actualizar esos modelos. se observa que la mejor frecuencia de actualización es para 0.95. (ver Tabla 4.5). Para observar la evaluación de cada una de las secuencias más en detalle. ver las tablas de los anexos (39. 40. 41. 42 y 43).

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
	Mean↑	Std↓	Mean↑	Std↓	Mean↓
$\varphi =$ sin tasa	0.5776	0.1637	0.4326	0.1434	0.8641
$\varphi = 0.85$	0.6164	0.1457	0.4684	0.1384	0.5924
$\varphi = 0.9$	0.6039	0.1425	0.4553	0.1385	0.6069
$\varphi = 0.95$	**0.6399**	**0.1185**	**0.4689**	**0.1247**	**0.4927**
$\varphi = 1$	0.5154	0.1495	0.3419	0.1364	0.3083

Tabla 4.5: Experimento 5: barrido del parámetro φ.

Como esquema general en la Figura 4.3 se muestran los parámetros escogidos que mejor parametrizan el algoritmo.

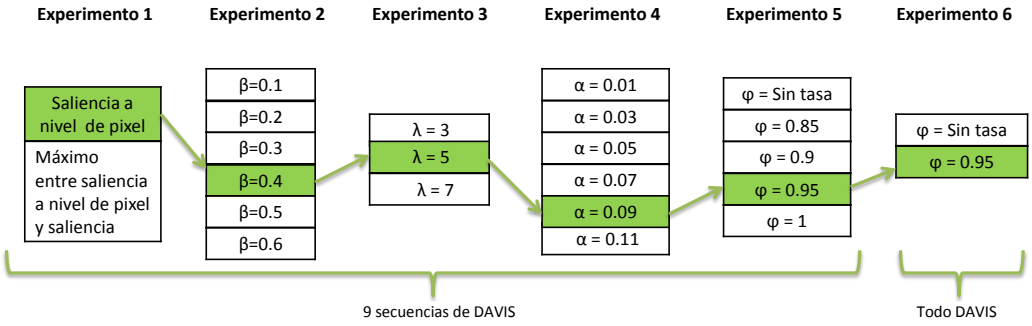


Figura 4.3: Este esquema muestra la mejor parametrización del algoritmo. Esta configuración se estima para las 9 secuencias de DAVIS. En el experimento 6 se realiza para todo DAVIS. salvo las 9 anteriores. empleando la misma parametrización que para las 9 secuencias anteriores.

4.2.6. Experimento 6: Evaluación de todo el *dataset*

El último experimento ha consistido en evaluar todo el *dataset*. salvo las 9 secuencias en las que se han realizado los anteriores experimentos. Los parámetros empleados han sido los mismos que se han conseguido en los anteriores experimentos. Es necesario tener en cuenta que cada secuencia es diferente y la elección de los parámetros ha sido en función de las 9 secuencias. Los resultados son los de las tablas 4.6 y 4.7.

	$\mathcal{J}$		$\boldsymbol{F}$		$\boldsymbol{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Bear	0.4216	0.3439	0.3814	0.2362	0.9676
Bmx-bumps	0.2403	0.1814	0.2597	0.1732	1.1753
Bmx-trees	0.4048	0.1244	0.4515	0.1164	0.8962
Boat	0.2780	0.0959	0.2167	0.0765	0.8299
Camel	0.4032	0.2633	0.3178	0.1762	1.1654
Car-shadow	0.5944	0.1636	0.3544	0.1173	0.8415
Car-turn	0.5561	0.1047	0.3632	0.0645	0.5478
Cows	0.2930	0.2118	0.1798	0.1290	1.1798
Dance-jump	0.360	0.1570	0.2544	0.1000	1.2607
Dance-twirl	0.3458	0.1496	0.2604	0.1099	1.2393
Dog-agility	0.2936	0.1945	0.1793	0.0978	1.7552
Drift-chicane	0.4555	0.2566	0.4732	0.2203	0.8518
Drift-straight	0.4852	0.1912	0.2959	0.2563	0.7507
Drift-turn	0.4522	0.2466	0.3312	0.2915	0.645
Goat	0.2246	0.1740	0.2122	0.1077	1.5779
Hike	0.7679	0.0902	0.7816	0.1155	0.2916
Hockey	0.4309	0.1526	0.3947	0.1148	0.6302
Horsejump-high	0.5679	0.0694	0.5538	0.0985	0.6956
Horsejump-low	0.5065	0.0915	0.4187	0.0778	0.8520
Kite-surf	0.2376	0.0990	0.2030	0.0728	1.1945
Kite-walk	0.5431	0.1117	0.4063	0.0804	0.6347
Libby	0.3453	0.1814	0.3798	0.1399	1.3924
Mallard-fly	0.3978	0.2470	0.3490	0.2289	1.2241
Mallard-water	0.0072	0.0178	0.0410	0.0648	1.6482
Motocross-bumps	0.3625	0.2118	0.3242	0.2111	0.8908
Motocross-jump	0.3544	0.2325	0.2712	0.1621	1.0669
Motorbike	0.5723	0.1573	0.4949	0.2027	0.7548
Paragliding	0.6939	0.0656	0.5856	0.1561	0.2245
Paragliding-lunch	0.4906	0.1175	0.1220	0.0552	0.3943
Parkour	0.4797	0.1401	0.4067	0.1451	0.8791
Rhino	0.2895	0.2899	0.2886	0.1872	1.2536
Rollerblade	0.3096	0.1394	0.2548	0.0867	1.0122
Scooter-black	0.4195	0.2990	0.3562	0.2166	1.0901
Scooter-gray	0.5208	0.1958	0.3249	0.1259	0.8702
Soapbox	0.3924	0.1615	0.3431	0.1407	0.8926
Surf	0.4213	0.2216	0.2278	0.1553	1.1052
Tennis	0.4800	0.1278	0.4483	0.1654	0.9304
Media	**0.4462**	**0.1697**	**0.3680**	**0.1426**	**0.9625**

Tabla 4.6: Experimento 6: Resultados evaluando DAVIS sin tasa de olvido(exceptuando las 9 secuencias anteriores).

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Bear	0.6550	0.3323	0.5913	0.2510	0.3317
Bmx-bumps	0.2551	0.2005	0.2425	0.2130	0.9252
Bmx-trees	0.4977	0.1163	0.4742	0.1214	0.4971
Boat	0.2771	0.1130	0.2180	0.0951	0.4553
Breakdance-flare	0.3908	0.1385	0.3726	0.1281	1.1935
Camel	0.5197	0.2169	0.4028	0.1429	0.4672
Car-shadow	0.7190	0.0986	0.4899	0.1750	0.3716
Car-turn	0.5769	0.0739	0.3510	0.0850	0.3313
Cows	0.4171	0.2113	0.3517	0.1462	0.5396
Dance-jump	0.4330	0.1076	0.3856	0.0870	0.5082
Dance-twirl	0.4041	0.1301	0.2893	0.1221	0.6871
Dog-agility	0.3215	0.1831	0.1785	0.0886	0.9410
Drift-chicane	0.6436	0.1698	0.6718	0.1980	0.3883
Drift-straight	0.5021	0.1704	0.3088	0.2351	0.5532
Drift-turn	0.4908	0.2158	0.3467	0.2748	0.4476
Elephant	0.3620	0.2629	0.23829	0.1469	0.7260
Goat	0.3556	0.1648	0.3385	0.1167	0.5595
Hike	0.7915	0.0617	0.8156	0.0965	0.1885
Hockey	0.4601	0.1542	0.4335	0.1145	0.4540
Horsejump-high	0.5881	0.0553	0.5592	0.0750	0.4860
Horsejump-low	0.5318	0.0639	0.4384	0.0789	0.5329
Kite-surf	0.2374	0.1227	0.4035	0.0759	0.6875
Kite-walk	0.5635	0.0852	0.4180	0.0765	0.4542
Libby	0.4382	0.1532	0.4395	0.1192	0.6967
Mallard-fly	0.4921	0.2427	0.4486	0.2440	0.8179
Mallard-water	0.0072	0.0178	0.0410	0.0648	1.6482
Motocross-bumps	0.3842	0.2142	0.3592	0.2338	0.5272
Motocross-jump	0.4299	0.2079	0.3959	0.1560	0.6928
Motorbike	0.6355	0.1196	0.5357	0.1890	0.4430
Paragliding	0.7221	0.0725	0.6232	0.1501	0.2057
Paragliding-lunch	0.5003	0.1131	0.4255	0.0594	0.3164
Parkour	0.5063	0.1421	0.4338	0.1436	0.6618
Rhino	0.4091	0.3198	0.3694	0.2220	0.6220
Rollerblade	0.5623	0.1365	0.4905	0.0925	0.6608
Scooter-black	0.4954	0.2075	0.3813	0.1658	0.6683
Scooter-gray	0.5605	0.1509	0.3390	0.1100	0.5101
Soapbox	0.5431	0.1596	0.4921	0.1236	0.5266
Stroller	0.4918	0.1008	0.3404	0.1064	0.5809
Surf	0.5016	0.2245	0.4194	0.1883	0.6770
Swing	0.5528	0.1104	0.3831	0.1003	0.6585
Tennis	0.5204	0.1387	0.4935	0.1788	0.6778
Media	**0.5116**	**0.1532**	**0.4768**	**0.1413**	**0.6231**

Tabla 4.7: Experimento 6: Resultados evaluando DAVIS con tasa de olvido 0.95(exceptuando las 9 secuencias anteriores)

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Algoritmos	Mean↑	Std↓	Mean↑	Std↓	Mean↓
CUT[12]	0.5421	0.1171	0.5354	0.1075	0.3874
CVOS[22]	0.4757	0.1068	0.5411	0.1098	0.4016
FST[18]	0.5815	0.1251	0.5411	0.1096	0.4247
KEY[14]	0.5767	0.1172	0.5159	0.1045	0.3302
MSG[17]	0.5441	0.1188	0.5312	0.1246	0.3846
NLC[4]	0.6413	0.0910	0.5931	0.0972	0.4756
TRC[6]	0.4967	0.1352	0.4815	0.1315	0.4624
Propuesto	0.5116	0.1532	0.4768	0.1413	0.6231

Tabla 4.8: Comparativa de los resultados del algoritmo implementado con ejemplos de algoritmos no supervisados

4.3. Comparativa de resultados

En esta sección se hace una comparativa del algoritmo implementado con otros algoritmos de segmentación no supervisados [12][22][18][14][17] [4][6]. De este análisis se puede deducir que la elección de los parámetrosα, β, λ y φ es un factor clave para obtener una buena segmentación. Esta configuración de los parámetros se ha extraido en base a 9 secuencias y se ha empleado para el resto de secuencias de DAVIS. Cada secuencia del *dataset* es distinta. algunas tienen mayor movimiento. otras en cambio oclusiones. sombras. etc y por ello en algunas secuencias quizá hubiese sido mejor otra parametrización. En base a este hecho . y la vista de los resultados de la Tabla 4.8. con el algoritmo implementado se consiguen unos resultados similares a otros algoritmos no supervisados. Con respecto a la región de similitud $\mathcal{J}$ se observa que en media se obtienen valores muy similares a los demás algoritmos. mientras que para la varianza se obtienen resultados un poco peores al resto. Para el caso de la precisión del contorno $\mathcal{F}$ se consiguen resultados bastante similares al resto con respecto a la media y para la varianza se consiguen resultados mejores que en algunos de los algoritmos del estado del arte.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

En este TFG se ha desarrollado un algoritmo capaz de segmentar de forma automática los objetos de interés en vídeos grabados con cámaras en movimiento. Para desarrollar este algoritmo se han utilizado técnicas del estado del arte que permiten identificar patrones espacio-temporales relevantes para segmentarlos.

En primer lugar se realizó un estudio del estado del arte para conocer las técnicas existentes en la literatura de este campo y se han analizado los diferentes tipos de tareas de segmentación frente-fondo y como se puede abordar dicha segmentación en cámaras en movimiento.

Tras estudiar el estado del arte, se ha desarrollar el algoritmo, que se puede clasificar en tres etapas. En la primera se ha obtenido una segmentación inicial o burda, para ello se han aplicado técnicas de flujo óptico y saliencia para estimar un modelo de movimiento y localización respectivamente. En esta primera etapa se ha comprobado que los vídeos donde los objetos carecen de movimiento, no se obtiene saliencia en movimiento y por tanto el modelo de localización no será robusto. Es por esto por lo que los resultados de la primera segmentación no pueden ser los finales ya que con el modelo de localización no se obtienen buenos resultados en la segmentación. La segunda etapa ha consistido en obtener una segunda segmentación con el modelo de localización previamente calculado y con un modelo de apariencia que se estima mediante modelos de mezclas de Gaussianas y un campo condicional aleatorio. De aquí se ha deducido que la elección del número de gaussianas que modelan el sistema es un factor clave para segmentar. Si se modela con un bajo número de gaussianas puede ocasionar la pérdida de información mientras que se modela con uno alto, el tiempo de ejecución será elevado. Por estas razones es importante elegir un número

que equilibre ambas características. Finalmente, en la última etapa del desarrollo, se ha calculado una segmentación final en la que se estima la calidad del modelo actual y se compara con el mejor modelo que se haya obtenido anteriormente, actualizándose cada vez que se tenga una mejor calidad. De esta etapa se ha comprobado que la actualización de modelos en el tiempo es fundamental, ya que en repetidas se realiza la segmentación con un modelo que no es el más aparente y por consiguiente , guardar los mejores modelos y emplearlos para segmentar en otros instantes permite que la segmentación sea más refinada.

Tras el desarrollo del algoritmo y con el objetivo de conseguir una parametrización adecuada del mismo, se realizaron 6 experimentos. En cada uno se modificaron diferentes parámetros con el fin de obtener la segmentación óptima. Dicha parametrización se ha estudiado sobre un conjunto de secuencias distinto al que se evaluó para garantizar que no existe un sobre ajuste. Las métricas de evaluación empleadas han sido variadas, midiendo la precisión del contorno, la similitud de las regiones y la estabilidad temporal del objeto segmentado en cada vídeo. Además, se ha comparado el sistema propuesto con otros algoritmos de segmentación no supervisada del estado del arte.

5.2. Trabajo futuro

A la vista de los resultados, la tarea de segmentación frente-fondo tiene un gran margen de mejora ya que ningún algoritmo funciona de manera precisa. La estimación de los objetos en movimiento con este algoritmo no es del todo precisa en las secuencias donde los objetos carecen de movimiento. Como trabajo futuro se pueden experimentar otro tipo de técnicas en las que se detecte mejor el movimiento para obtener un modelo de localización más robusto.

En la segmentación final, se ha tratado de conseguir un refinamiento del modelo, para estimar mejores modelos se podrían utilizar características y modelos más complejos como los que proporcionan las redes neuronales actuales y las redes neuronales convolucionales.

En la evaluación del algoritmo se ha comprobado que dependiendo de la secuencia del *datset* que se esté analizando, el umbral es un factor clave para clasificar los píxeles como máscara de frente o fondo. Para mejorar en este aspecto, se propone como trabajo futuro estimar un umbral adaptativo que se ajuste a las diferentes secuencias de vídeo que se estén evaluando.

Bibliografía

[1] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A benchmark. *IEEE Transactions on Image Processing*, 24:5706–5722, 2015.

[2] T. Bouwmans. Traditional and recent approaches in background modeling for foreground detection: An overview. *Computer Science Review*, 11-12:31–66, 2014.

[3] William Brendel and Sinisa Todorovic. Video object segmentation by tracking regions. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 833–840, 2009.

[4] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *Proceedings of British Machine Vision Conference (BMVC)*, 2014.

[5] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.

[6] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012.

[7] Daniela Giordano, Francesca Murabito, Simone Palazzo, and Concetto Spampinato. Superpixel-based video object segmentation using perceptual organization and location prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[8] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, pages 545–552. 2007.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2017.

[10] Diego Ortego Hernandez. *Quality-driven video analysis for the improvemet of foreground segmentation.* PhD thesis, Universidad Autonoma de Madrid, May 2018.

[11] Won-Dong Jang and Chang-Su Kim. Online video object segmentation via convolutional trident network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[12] Margret Keuper, Bjoern Andres, and Thomas Brox. Motion trajectory segmentation via minimum cost multicuts. In *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015.

[13] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. *arXiv preprint arXiv:1801.00868*, 2018.

[14] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.

[15] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. *arXiv preprint arXiv:1711.09081*, 2017.

[16] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 4293–4302. IEEE, 2016.

[17] Peter Ochs and Thomas Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011.

[18] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013.

[19] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[20] Federico Perazzi, Oliver Wang, Markus Gross, and Alexander Sorkine-Hornung. Fully connected object proposals for video segmentation. In *Proceedings of the IEEE international conference on computer vision*, 2015.

[21] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[22] Brian Taylor, Vasiliy Karasev, and Stefano Soatto. Causal video object segmentation from persistence of occlusions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[23] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[24] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[25] Wenguan Wang, Jianbing Shen, and Ling Shao. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE Transactions on Image Processing*, 24(11):4185–4196, 2015.

[26] Chenliang Xu and Jason J Corso. Actor-action semantic segmentation with grouping process models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3083–3092, 2016.

[27] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[Experimento1: Elección del máximo entre la saliencia a nivel de superpixel o saliencia]Experimento1: Elección del máximo entre la saliencia a nivel de superpixel o saliencia. Resultados obtenidos pasra 9 secuencias de DAVIS

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	**Mean↑**	**Std↓**	**Mean↑**	**Std↓**	**Mean↓**
Blackswan	0.3670	0.1006	0.3799	0.1316	1.1769
Breakdance	0.2270	0.1492	0.1885	0.0961	1.4005
Bus	0.8038	0.1140	0.4741	0.1520	0.2332
Car-roundabout	0.6390	0.1781	0.3828	0.1550	0.6950
Dog	0.4142	0.2641	0.2821	0.1696	1.2014
Flamingo	0.4967	0.2272	0.5017	0.1595	0.8598
Lucia	0.6699	0.1165	0.5450	0.1181	0.4302
Soccerball	0.5628	0.1271	0.4934	0.1188	0.4642
Train	0.6639	0.1575	0.4372	0.1323	0.5657
Media	0.5382	0.1594	0.4094	0.1370	0.7808

Tabla 1: Experimento 1: Resultados obtenidos para 9 secuencias de DAVIS

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3450	0.0816	0.2685	0.1215	1.1198
Breakdance	0.2046	0.1486	0.1735	0.0896	1.4183
Bus	0.7180	0.1620	0.3722	0.1314	0.5803
Car-roundabout	0.6395	0.1853	0.3794	0.1503	0.7247
Dog	0.4479	0.2276	0.3018	0.1538	1.3400
Flamingo	0.5372	0.1710	0.4907	0.1551	0.9161
Lucia	0.6116	0.1137	0.4765	0.1190	0.4719
Soccerball	0.5578	0.1427	0.4905	0.136	0.4856
Train	0.6037	0.1803	0.4138	0.1457	0.6133
Media	**0.5184**	**0.1570**	**0.3741**	**0.1336**	**0.9530**

Tabla 2: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.1.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3501	0.0963	0.3206	0.1475	1.2324
Breakdance	0.2407	0.1615	0.2022	0.1001	1.4064
Bus	0.8111	0.0868	0.4724	0.1319	0.1679
Car-roundabout	0.6403	0.1721	0.3531	0.1465	0.6037
Dog	0.4388	0.2533	0.2924	0.1704	1.3898
Flamingo	0.5555	0.215	0.5551	0.1483	0.8582
Lucia	0.6729	0.07914	0.5407	0.1118	0.3682
Soccerball	0.5734	0.1344	0.5013	0.1279	0.4564
Train	0.6351	0.1728	0.4279	0.1371	0.6401
Media	**0.5464**	**0.1524**	**0.4073**	**0.1357**	**0.7915**

Tabla 4: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.3.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3573	0.1195	0.3822	0.1356	1.0520
Breakdance	0.2227	0.1435	0.1897	0.0960	1.4379
Bus	0.7979	0.1155	0.4688	0.1451	0.3005
Car-roundabout	0.6304	0.1832	0.3637	0.1546	0.7072
Dog	0.4015	0.2575	0.2656	0.1565	1.2558
Flamingo	0.4478	0.2242	0.4512	0.1613	0.8027
Lucia	0.6611	0.1190	0.5419	0.1150	0.4720
Soccerball	0.5565	0.1316	0.4913	0.1252	0.4851
Train	0.634	0.1719	0.3985	0.1274	0.6588
Media	**0.5235**	**0.1629**	**0.3948**	**0.1352**	**0.7969**

Tabla 6: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.5.

	J		***F***		***T***
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3392	0.1557	0.3408	0.1751	1.0987
Breakdance	0.2200	0.1344	0.1918	0.0929	1.5532
Bus	0.7930	0.1319	0.4613	0.1420	0.2892
Car-roundabout	0.6236	0.1792	0.3603	0.1595	0.8034
Dog	0.1340	0.2546	0.0891	0.1535	0.7083
Flamingo	0.4206	0.2306	0.4306	0.1540	0.9789
Lucia	0.6363	0.1467	0.5201	0.1187	0.5687
Soccerball	0.5271	0.1580	0.4674	0.1471	0.5697
Train	0.6202	0.1561	0.3934	0.1219	0.6976
Media	**0.4793**	**0.1719**	**0.3616**	**0.1405**	**0.8075**

Tabla 7: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.6.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3670	0.1006	0.3799	0.1316	1.1769
Breakdance	0.2270	0.1492	0.1885	0.0961	1.4005
Bus	0.8051	0.1125	0.4722	0.1482	0.2426
Car-roundabout	0.6390	0.1781	0.3828	0.1550	0.6950
Dog	0.4142	0.2641	0.2821	0.1696	1.2014
flamingo	0.4967	0.2272	0.5017	0.1595	0.8598
lucia	0.6699	0.1165	0.5450	0.1181	0.4302
soccerball	0.5628	0.1271	0.4934	0.1188	0.4642
train	0.6639	0.1575	0.4372	0.1323	0.5657
Media	**0.5384**	**0.1592**	**0.4092**	**0.1366**	**0.7818**

Tabla 8: Experimento 3: Resultados obtenidos para 9 secuencias de DAVIS modelando con 3 gaussianas.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3407	0.1085	0.3672	0.1419	1.1161
Breakdance	0.2245	0.1472	0.1862	0.0958	1.4525
Bus	0.8030	0.1131	0.4784	0.1461	0.2751
Car-roundabout	0.6446	0.1818	0.3739	0.1604	0.6534
Dog	0.43118	0.2508	0.2916	0.1700	1.2075
Flamingo	0.4423	0.2259	0.4783	0.1592	1.1186
Lucia	0.6617	0.1162	0.5336	0.1291	0.4774
Soccerball	0.5680	0.1344	0.5042	0.1305	0.4456
Train	0.6429	0.1534	0.3851	0.1555	0.7123
Media	**0.5688**	**0.1590**	**0.4198**	**0.1432**	**0.8487**

Tabla 9: Experimento 3: Resultados obtenidos para 9 secuencias de DAVIS modelando con 5 gaussianas.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3600	0.1043	0.3831	0.1413	1.1455
Breakdance	0.2337	0.1542	0.1965	0.1003	1.4628
Bus	0.7906	0.1182	0.4702	0.1470	0.4019
Car-roundabout	0.6323	0.1857	0.3722	0.1570	0.6761
Dog	0.4314	0.2491	0.2893	0.1641	1.3082
Flamingo	0.4683	0.2273	0.4924	0.1576	0.9513
Lucia	0.6669	0.1130	0.5516	0.1181	0.4866
Soccerball	0.5685	0.1343	0.4927	0.1202	0.4546
Train	0.6413	0.1525	0.3535	0.1472	0.6501
Media	**0.5726**	**0.1598**	**0.4202**	**0.1392**	**0.8574**

Tabla 10: Experimento 3: Resultados obtenidos para 9 secuencias de DAVIS modelando con 7 gaussianas

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.2807	0.0835	0.2253	0.0913	0.90140
Breakdance	0.2150	0.1443	0.1821	0.0944	1.3983
Bus	0.7884	0.1029	0.4419	0.1275	0.3110
Car-roundabout	0.5968	0.1868	0.2850	0.1673	0.6742
Dog	0.5083	0.2056	0.2900	0.1417	0.8852
Flamingo	0.4698	0.2042	0.4511	0.1438	1.0684
Lucia	0.5989	0.1047	0.3980	0.1045	0.4430
Soccerball	0.5782	0.1327	0.4997	0.1388	0.4254
Train	0.6162	0.1426	0.2868	0.1420	0.6624
Media	**0.5569**	**0.1453**	**0.3600**	**0.1279**	**0.7721**

Tabla 11: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.01.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3161	0.0826	0.2835	0.1077	0.9185
Breakdance	0.2240	0.1522	0.1893	0.1048	1.4358
Bus	0.8029	0.1031	0.4648	0.1320	0.2551
Car-roundabout	0.6239	0.1763	0.3341	0.157	0.6641
Dog	0.4813	0.2321	0.3033	0.1658	1.2374
Flamingo	0.4641	0.2100	0.4723	0.1423	1.0529
Lucia	0.6398	0.1151	0.4742	0.1149	0.4575
Soccerball	0.5723	0.1267	0.4907	0.1272	0.4205
Train	0.62698	0.1463	0.3082	0.1457	0.7438
Media	**0.5679**	**0.1494**	**0.3890**	**0.1331**	**0.8184**

Tabla 12: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.03.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3493	0.0766	0.3596	0.1100	0.9251
Breakdance	0.2241	0.1529	0.1886	0.0938	1.4538
Bus	0.7892	0.1254	0.4691	0.1453	0.3768
Car-roundabout	0.6369	0.1809	0.3553	0.1600	0.6317
Dog	0.4455	0.2514	0.3072	0.1664	1.1933
Flamingo	0.4624	0.2321	0.4726	0.1659	0.9397
Lucia	0.6528	0.1109	0.5226	0.1207	0.5141
Soccerball	0.5850	0.1368	0.5059	0.1382	0.4343
Train	0.6380	0.1434	0.3613	0.1577	0.7162
Media	**0.5715**	**0.1567**	**0.4136**	**0.1398**	**0.8183**

Tabla 13: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.05

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3600	0.1043	0.3831	0.1413	1.1455
Breakdance	0.2337	0.1542	0.1965	0.1003	1.4628
Bus	0.7906	0.11824	0.4702	0.1470	0.4019
Car-roundabout	0.6323	0.1857	0.3722	0.1570	0.6761
Dog	0.4314	0.2491	0.2893	0.1641	1.3082
flamingo	0.4683	0.2273	0.4924	0.1576	0.9513
lucia	0.6669	0.1130	0.5516	0.1181	0.4866
soccerball	0.5685	0.1343	0.4927	0.1202	0.4546
train	0.6413	0.1525	0.3535	0.1472	0.6501
Media	**0.57260**	**0.1598**	**0.4202**	**0.1392**	**0.8574**

Tabla 14: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.07

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3651	0.1150	0.3770	0.1396	1.2928
Bbreakdance	0.2415	0.1464	0.2042	0.1008	1.4374
Bus	0.8031	0.1081	0.4867	0.1344	0.3238
Car-roundabout	0.6725	0.1725	0.4131	0.1605	0.5861
Dog	0.4392	0.2621	0.3123	0.1693	1.2205
Flamingo	0.4481	0.2349	0.4757	0.1698	0.9802
Lucia	0.6617	0.1202	0.5539	0.1199	0.5516
Soccerball	0.5665	0.1442	0.4935	0.1367	0.4569
Train	0.6409	0.1696	0.3966	0.1602	0.7476
Media	**0.5776**	**0.1637**	**0.4326**	**0.1434**	**0.8641**

Tabla 15: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.09.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3836	0.1395	0.3727	0.1439	1.4365
Breakdance	0.2468	0.1516	0.2015	0.0983	1.4667
Bus	0.8107	0.1030	0.4920	0.1395	0.2676
Car-roundabout	0.6643	0.1842	0.4007	0.1620	0.5989
Dog	0.4007	0.2846	0.2928	0.1955	1.4120
Flamingo	0.4415	0.2233	0.4716	0.1587	0.9520
Lucia	0.6710	0.1156	0.5693	0.1179	0.4791
Soccerball	0.5630	0.1421	0.4879	0.1255	0.5107
Train	0.6440	0.1612	0.3955	0.1632	0.6572
Media	**0.5762**	**0.1672**	**0.4293**	**0.1449**	**0.884**

Tabla 16: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.11.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.4195	0.1153	0.3970	0.1330	0.7920
Breakdance	0.2728	0.1572	0.2180	0.1098	1.0775
Bus	0.8178	0.1024	0.5071	0.1474	0.3088
Dog	0.4871	0.2586	0.3767	0.1877	0.942
Car-roundabout	0.6861	0.1411	0.4063	0.1712	0.4085
Flamingo	0.4971	0.2019	0.5102	0.1178	0.5367
Lucia	0.7051	0.0794	0.6052	0.1040	0.2886
Soccerball	0.6105	0.1453	0.5646	0.1378	0.5073
Train	0.6916	0.1099	0.4505	0.1373	0.2896
Media	**0.6164**	**0.1457**	**0.4684**	**0.1384**	**0.5924**

Tabla 18: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS con tasa de olvido 0.85.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3980	0.1060	0.3711	0.1198	0.7588
Breakdance	0.2903	0.1479	0.2270	0.1099	1.0879
Bus	0.8002	0.1063	0.4848	0.1500	0.2579
Dog	0.4871	0.2586	0.3767	0.1877	0.9421
Car-roundabout	0.6909	0.1199	0.4041	0.1737	0.3410
Flamingo	0.4361	0.1781	0.4850	0.1106	0.7147
Lucia	0.696	0.0851	0.5976	0.1062	0.3330
Soccerball	0.5965	0.1584	0.5393	0.1539	0.4827
Train	0.6788	0.1217	0.4320	0.1347	0.3643
Media	**0.6039**	**0.1425**	**0.4553**	**0.1385**	**0.6069**

Tabla 19: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS con tasa de olvido 0.9.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.4387	0.1214	0.3665	0.1320	0.6769
Breakdance	0.2949	0.1532	0.2378	0.1135	0.9255
Bus	0.8194	0.0975	0.5005	0.14707	0.2147
Dog	0.5782	0.1827	0.4243	0.1548	0.7372
Car-roundabout	0.7106	0.1074	0.4255	0.1651	0.3403
Flamingo	0.5308	0.0885	0.4325	0.0507	0.2333
Lucia	0.7188	0.0629	0.6226	0.0891	0.2810
Soccerball	0.6105	0.1453	0.5646	0.1378	0.5073
Train	0.6976	0.1077	0.4658	0.1325	0.3381
Media	**0.6399**	**0.1185**	**0.4689**	**0.1247**	**0.4927**

Tabla 20: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS con tasa de olvido 0.95.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.4096	0.1395	0.2715	0.0960	0.3626
Breakdance	0.1432	0.1165	0.1372	0.0929	0.5055
Bus	0.7743	0.1352	0.5466	0.1123	0.1204
Dog	0.4002	0.2181	0.2399	0.1683	0.5759
Car-roundabout	0.6280	0.1203	0.3705	0.1330	0.1689
Flamingo	0.5308	0.0885	0.4325	0.0507	0.2333
Lucia	0.6229	0.0776	0.4317	0.1115	0.0795
Soccerball	0.1803	0.2761	0.1681	0.2535	0.5023
Train	0.5890	0.1734	0.2995	0.2092	0.0461
Media	**0.5154**	**0.1495**	**0.3419**	**0.1364**	**0.3083**

Tabla 21: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS con tasa de olvido 1.

]

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	**Mean↑**	**Std↓**	**Mean↑**	**Std↓**	**Mean↓**
Blackswan	0.2966	0.0835	0.2136	0.0824	0.7823
Breakdance	0.2411	0.1527	0.1944	0.0960	1.3797
Bus	0.7838	0.1159	0.4255	0.1283	0.2274
Car-roundabout	0.6212	0.1744	0.3161	0.1667	0.6570
Dog	0.4873	0.2114	0.3064	0.1416	1.0146
Flamingo	0.5407	0.2065	0.5343	0.1540	0.7761
Lucia	0.6179	0.0963	0.4185	0.1046	0.3944
Soccerball	0.5621	0.1319	0.4963	0.1251	0.4818
Train	0.6454	0.1613	0.3671	0.1359	0.6246
Media	0.5329	0.1482	0.3636	0.1261	0.7042

Tabla 22: Experimento1: Elección del máximo entre la saliencia a nivel de superpixel o saliencia. Resultados obtenidos pasra 9 secuencias de DAVIS

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	**Mean↑**	**Std↓**	**Mean↑**	**Std↓**	**Mean↓**
Blackswan	0.3670	0.1006	0.3799	0.1316	1.1769
Breakdance	0.2270	0.1492	0.1885	0.0961	1.4005
Bus	0.8038	0.1140	0.4741	0.1520	0.2332
Car-roundabout	0.6390	0.1781	0.3828	0.1550	0.6950
Dog	0.4142	0.2641	0.2821	0.1696	1.2014
Flamingo	0.4967	0.2272	0.5017	0.1595	0.8598
Lucia	0.6699	0.1165	0.5450	0.1181	0.4302
Soccerball	0.5628	0.1271	0.4934	0.1188	0.4642
Train	0.6639	0.1575	0.4372	0.1323	0.5657
Media	0.5382	0.1594	0.4094	0.1370	0.7808

Tabla 23: Experimento 1: Resultados obtenidos para 9 secuencias de DAVIS

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3450	0.0816	0.2685	0.1215	1.1198
Breakdance	0.2046	0.1486	0.1735	0.0896	1.4183
Bus	0.7180	0.1620	0.3722	0.1314	0.5803
Car-roundabout	0.6395	0.1853	0.3794	0.1503	0.7247
Dog	0.4479	0.2276	0.3018	0.1538	1.3400
Flamingo	0.5372	0.1710	0.4907	0.1551	0.9161
Lucia	0.6116	0.1137	0.4765	0.1190	0.4719
Soccerball	0.5578	0.1427	0.4905	0.136	0.4856
Train	0.6037	0.1803	0.4138	0.1457	0.6133
Media	**0.5184**	**0.1570**	**0.3741**	**0.1336**	**0.9530**

Tabla 24: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.1.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3501	0.0963	0.3206	0.1475	1.2324
Breakdance	0.2407	0.1615	0.2022	0.1001	1.4064
Bus	0.8111	0.0868	0.4724	0.1319	0.1679
Car-roundabout	0.6403	0.1721	0.3531	0.1465	0.6037
Dog	0.4388	0.2533	0.2924	0.1704	1.3898
Flamingo	0.5555	0.215	0.5551	0.1483	0.8582
Lucia	0.6729	0.07914	0.5407	0.1118	0.3682
Soccerball	0.5734	0.1344	0.5013	0.1279	0.4564
Train	0.6351	0.1728	0.4279	0.1371	0.6401
Media	**0.5464**	**0.1524**	**0.4073**	**0.1357**	**0.7915**

Tabla 26: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.3.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3573	0.1195	0.3822	0.1356	1.0520
Breakdance	0.2227	0.1435	0.1897	0.0960	1.4379
Bus	0.7979	0.1155	0.4688	0.1451	0.3005
Car-roundabout	0.6304	0.1832	0.3637	0.1546	0.7072
Dog	0.4015	0.2575	0.2656	0.1565	1.2558
Flamingo	0.4478	0.2242	0.4512	0.1613	0.8027
Lucia	0.6611	0.1190	0.5419	0.1150	0.4720
Soccerball	0.5565	0.1316	0.4913	0.1252	0.4851
Train	0.634	0.1719	0.3985	0.1274	0.6588
Media	**0.5235**	**0.1629**	**0.3948**	**0.1352**	**0.7969**

Tabla 28: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.5.

	J		***F***		***T***
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3392	0.1557	0.3408	0.1751	1.0987
Breakdance	0.2200	0.1344	0.1918	0.0929	1.5532
Bus	0.7930	0.1319	0.4613	0.1420	0.2892
Car-roundabout	0.6236	0.1792	0.3603	0.1595	0.8034
Dog	0.1340	0.2546	0.0891	0.1535	0.7083
Flamingo	0.4206	0.2306	0.4306	0.1540	0.9789
Lucia	0.6363	0.1467	0.5201	0.1187	0.5687
Soccerball	0.5271	0.1580	0.4674	0.1471	0.5697
Train	0.6202	0.1561	0.3934	0.1219	0.6976
Media	**0.4793**	**0.1719**	**0.3616**	**0.1405**	**0.8075**

Tabla 29: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.6.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3670	0.1006	0.3799	0.1316	1.1769
Breakdance	0.2270	0.1492	0.1885	0.0961	1.4005
Bus	0.8051	0.1125	0.4722	0.1482	0.2426
Car-roundabout	0.6390	0.1781	0.3828	0.1550	0.6950
Dog	0.4142	0.2641	0.2821	0.1696	1.2014
flamingo	0.4967	0.2272	0.5017	0.1595	0.8598
lucia	0.6699	0.1165	0.5450	0.1181	0.4302
soccerball	0.5628	0.1271	0.4934	0.1188	0.4642
train	0.6639	0.1575	0.4372	0.1323	0.5657
Media	**0.5384**	**0.1592**	**0.4092**	**0.1366**	**0.7818**

Tabla 30: Experimento 3: Resultados obtenidos para 9 secuencias de DAVIS modelando con 3 gaussianas.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3407	0.1085	0.3672	0.1419	1.1161
Breakdance	0.2245	0.1472	0.1862	0.0958	1.4525
Bus	0.8030	0.1131	0.4784	0.1461	0.2751
Car-roundabout	0.6446	0.1818	0.3739	0.1604	0.6534
Dog	0.43118	0.2508	0.2916	0.1700	1.2075
Flamingo	0.4423	0.2259	0.4783	0.1592	1.1186
Lucia	0.6617	0.1162	0.5336	0.1291	0.4774
Soccerball	0.5680	0.1344	0.5042	0.1305	0.4456
Train	0.6429	0.1534	0.3851	0.1555	0.7123
Media	**0.5688**	**0.1590**	**0.4198**	**0.1432**	**0.8487**

Tabla 31: Experimento 3: Resultados obtenidos para 9 secuencias de DAVIS modelando con 5 gaussianas.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3600	0.1043	0.3831	0.1413	1.1455
Breakdance	0.2337	0.1542	0.1965	0.1003	1.4628
Bus	0.7906	0.1182	0.4702	0.1470	0.4019
Car-roundabout	0.6323	0.1857	0.3722	0.1570	0.6761
Dog	0.4314	0.2491	0.2893	0.1641	1.3082
Flamingo	0.4683	0.2273	0.4924	0.1576	0.9513
Lucia	0.6669	0.1130	0.5516	0.1181	0.4866
Soccerball	0.5685	0.1343	0.4927	0.1202	0.4546
Train	0.6413	0.1525	0.3535	0.1472	0.6501
Media	**0.5726**	**0.1598**	**0.4202**	**0.1392**	**0.8574**

Tabla 32: Experimento 3: Resultados obtenidos para 9 secuencias de DAVIS modelando con 7 gaussianas

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.2807	0.0835	0.2253	0.0913	0.90140
Breakdance	0.2150	0.1443	0.1821	0.0944	1.3983
Bus	0.7884	0.1029	0.4419	0.1275	0.3110
Car-roundabout	0.5968	0.1868	0.2850	0.1673	0.6742
Dog	0.5083	0.2056	0.2900	0.1417	0.8852
Flamingo	0.4698	0.2042	0.4511	0.1438	1.0684
Lucia	0.5989	0.1047	0.3980	0.1045	0.4430
Soccerball	0.5782	0.1327	0.4997	0.1388	0.4254
Train	0.6162	0.1426	0.2868	0.1420	0.6624
Media	**0.5569**	**0.1453**	**0.3600**	**0.1279**	**0.7721**

Tabla 33: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.01.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3161	0.0826	0.2835	0.1077	0.9185
Breakdance	0.2240	0.1522	0.1893	0.1048	1.4358
Bus	0.8029	0.1031	0.4648	0.1320	0.2551
Car-roundabout	0.6239	0.1763	0.3341	0.157	0.6641
Dog	0.4813	0.2321	0.3033	0.1658	1.2374
Flamingo	0.4641	0.2100	0.4723	0.1423	1.0529
Lucia	0.6398	0.1151	0.4742	0.1149	0.4575
Soccerball	0.5723	0.1267	0.4907	0.1272	0.4205
Train	0.62698	0.1463	0.3082	0.1457	0.7438
Media	**0.5679**	**0.1494**	**0.3890**	**0.1331**	**0.8184**

Tabla 34: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.03.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3493	0.0766	0.3596	0.1100	0.9251
Breakdance	0.2241	0.1529	0.1886	0.0938	1.4538
Bus	0.7892	0.1254	0.4691	0.1453	0.3768
Car-roundabout	0.6369	0.1809	0.3553	0.1600	0.6317
Dog	0.4455	0.2514	0.3072	0.1664	1.1933
Flamingo	0.4624	0.2321	0.4726	0.1659	0.9397
Lucia	0.6528	0.1109	0.5226	0.1207	0.5141
Soccerball	0.5850	0.1368	0.5059	0.1382	0.4343
Train	0.6380	0.1434	0.3613	0.1577	0.7162
Media	**0.5715**	**0.1567**	**0.4136**	**0.1398**	**0.8183**

Tabla 35: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.05

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3600	0.1043	0.3831	0.1413	1.1455
Breakdance	0.2337	0.1542	0.1965	0.1003	1.4628
Bus	0.7906	0.11824	0.4702	0.1470	0.4019
Car-roundabout	0.6323	0.1857	0.3722	0.1570	0.6761
Dog	0.4314	0.2491	0.2893	0.1641	1.3082
flamingo	0.4683	0.2273	0.4924	0.1576	0.9513
lucia	0.6669	0.1130	0.5516	0.1181	0.4866
soccerball	0.5685	0.1343	0.4927	0.1202	0.4546
train	0.6413	0.1525	0.3535	0.1472	0.6501
Media	**0.57260**	**0.1598**	**0.4202**	**0.1392**	**0.8574**

Tabla 36: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.07

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3651	0.1150	0.3770	0.1396	1.2928
Bbreakdance	0.2415	0.1464	0.2042	0.1008	1.4374
Bus	0.8031	0.1081	0.4867	0.1344	0.3238
Car-roundabout	0.6725	0.1725	0.4131	0.1605	0.5861
Dog	0.4392	0.2621	0.3123	0.1693	1.2205
Flamingo	0.4481	0.2349	0.4757	0.1698	0.9802
Lucia	0.6617	0.1202	0.5539	0.1199	0.5516
Soccerball	0.5665	0.1442	0.4935	0.1367	0.4569
Train	0.6409	0.1696	0.3966	0.1602	0.7476
Media	**0.5776**	**0.1637**	**0.4326**	**0.1434**	**0.8641**

Tabla 37: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.09.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3836	0.1395	0.3727	0.1439	1.4365
Breakdance	0.2468	0.1516	0.2015	0.0983	1.4667
Bus	0.8107	0.1030	0.4920	0.1395	0.2676
Car-roundabout	0.6643	0.1842	0.4007	0.1620	0.5989
Dog	0.4007	0.2846	0.2928	0.1955	1.4120
Flamingo	0.4415	0.2233	0.4716	0.1587	0.9520
Lucia	0.6710	0.1156	0.5693	0.1179	0.4791
Soccerball	0.5630	0.1421	0.4879	0.1255	0.5107
Train	0.6440	0.1612	0.3955	0.1632	0.6572
Media	**0.5762**	**0.1672**	**0.4293**	**0.1449**	**0.884**

Tabla 38: Experimento 4: Resultados obtenidos para 9 secuencias de DAVIS con un parámetro de merging de 0.11.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.4195	0.1153	0.3970	0.1330	0.7920
Breakdance	0.2728	0.1572	0.2180	0.1098	1.0775
Bus	0.8178	0.1024	0.5071	0.1474	0.3088
Dog	0.4871	0.2586	0.3767	0.1877	0.942
Car-roundabout	0.6861	0.1411	0.4063	0.1712	0.4085
Flamingo	0.4971	0.2019	0.5102	0.1178	0.5367
Lucia	0.7051	0.0794	0.6052	0.1040	0.2886
Soccerball	0.6105	0.1453	0.5646	0.1378	0.5073
Train	0.6916	0.1099	0.4505	0.1373	0.2896
Media	**0.6164**	**0.1457**	**0.4684**	**0.1384**	**0.5924**

Tabla 40: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS con tasa de olvido 0.85.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3980	0.1060	0.3711	0.1198	0.7588
Breakdance	0.2903	0.1479	0.2270	0.1099	1.0879
Bus	0.8002	0.1063	0.4848	0.1500	0.2579
Dog	0.4871	0.2586	0.3767	0.1877	0.9421
Car-roundabout	0.6909	0.1199	0.4041	0.1737	0.3410
Flamingo	0.4361	0.1781	0.4850	0.1106	0.7147
Lucia	0.696	0.0851	0.5976	0.1062	0.3330
Soccerball	0.5965	0.1584	0.5393	0.1539	0.4827
Train	0.6788	0.1217	0.4320	0.1347	0.3643
Media	**0.6039**	**0.1425**	**0.4553**	**0.1385**	**0.6069**

Tabla 41: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS con tasa de olvido 0.9.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.4387	0.1214	0.3665	0.1320	0.6769
Breakdance	0.2949	0.1532	0.2378	0.1135	0.9255
Bus	0.8194	0.0975	0.5005	0.14707	0.2147
Dog	0.5782	0.1827	0.4243	0.1548	0.7372
Car-roundabout	0.7106	0.1074	0.4255	0.1651	0.3403
Flamingo	0.5308	0.0885	0.4325	0.0507	0.2333
Lucia	0.7188	0.0629	0.6226	0.0891	0.2810
Soccerball	0.6105	0.1453	0.5646	0.1378	0.5073
Train	0.6976	0.1077	0.4658	0.1325	0.3381
Media	**0.6399**	**0.1185**	**0.4689**	**0.1247**	**0.4927**

Tabla 42: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS con tasa de olvido 0.95.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.4096	0.1395	0.2715	0.0960	0.3626
Breakdance	0.1432	0.1165	0.1372	0.0929	0.5055
Bus	0.7743	0.1352	0.5466	0.1123	0.1204
Dog	0.4002	0.2181	0.2399	0.1683	0.5759
Car-roundabout	0.6280	0.1203	0.3705	0.1330	0.1689
Flamingo	0.5308	0.0885	0.4325	0.0507	0.2333
Lucia	0.6229	0.0776	0.4317	0.1115	0.0795
Soccerball	0.1803	0.2761	0.1681	0.2535	0.5023
Train	0.5890	0.1734	0.2995	0.2092	0.0461
Media	**0.5154**	**0.1495**	**0.3419**	**0.1364**	**0.3083**

Tabla 43: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS con tasa de olvido 1.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3636	0.0837	0.2722	0.1228	0.9305
Breakdance	0.2245	0.1508	0.1847	0.0938	1.4770
Bus	0.8137	0.0807	0.4663	0.1282	0.1921
Car-roundabout	0.6310	0.1578	0.34644	0.1304	0.7551
Dog	0.4384	0.2319	0.2983	0.1608	1.3259
Flamingo	0.5922	0.1666	0.5720	0.1146	0.8276
Lucia	0.6522	0.1104	0.5250	0.1225	0.4564
Soccerball	0.5775	0.1401	0.5103	0.1310	0.4383
Train	0.6092	0.1677	0.4235	0.1442	0.8609
Media	0.5447	0.1433	0.3998	0.1276	0.8071

Tabla 3: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.2

	J		***F***		***T***
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3670	0.1006	0.3799	0.1316	1.1769
Breakdance	0.2270	0.1492	0.1885	0.0961	1.4005
Bus	0.8038	0.1140	0.4741	0.152	0.2332
Car-roundabout	0.6390	0.1781	0.3828	0.1550	0.6950
Dog	0.4142	0.2641	0.2821	0.1696	1.2014
Flamingo	0.4967	0.2272	0.5017	0.1595	0.8598
Lucia	0.6699	0.1165	0.5450	0.1181	0.4302
Soccerball	0.5628	0.1271	0.4934	0.1188	0.4642
Train	0.6639	0.1575	0.4372	0.1323	0.5657
Media	**0.5382**	**0.1594**	**0.4094**	**0.1370**	**0.7808**

Tabla 5: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.4.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3651	0.1150	0.3770	0.1396	1.2928
Breakdance	0.2415	0.1464	0.2042	0.1008	1.4374
Bus	0.8031	0.1081	0.4867	0.1344	0.3238
Car-roundabout	0.6725	0.1725	0.4131	0.1605	0.5861
Dog	0.4392	0.2621	0.3123	0.1693	1.2205
Flamingo	0.4481	0.2349	0.4757	0.1698	0.9802
Lucia	0.6617	0.1202	0.5539	0.1199	0.5516
Soccerball	0.5665	0.1442	0.4935	0.1367	0.4569
Train	0.6409	0.1696	0.3966	0.1602	0.7476
Media	**0.5776**	**0.1637**	**0.4326**	**0.1434**	**0.8641**

Tabla 17: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS sin tasa de olvido

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3636	0.0837	0.2722	0.1228	0.9305
Breakdance	0.2245	0.1508	0.1847	0.0938	1.4770
Bus	0.8137	0.0807	0.4663	0.1282	0.1921
Car-roundabout	0.6310	0.1578	0.34644	0.1304	0.7551
Dog	0.4384	0.2319	0.2983	0.1608	1.3259
Flamingo	0.5922	0.1666	0.5720	0.1146	0.8276
Lucia	0.6522	0.1104	0.5250	0.1225	0.4564
Soccerball	0.5775	0.1401	0.5103	0.1310	0.4383
Train	0.6092	0.1677	0.4235	0.1442	0.8609
Media	0.5447	0.1433	0.3998	0.1276	0.8071

Tabla 25: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.2

	J		***F***		***T***
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3670	0.1006	0.3799	0.1316	1.1769
Breakdance	0.2270	0.1492	0.1885	0.0961	1.4005
Bus	0.8038	0.1140	0.4741	0.152	0.2332
Car-roundabout	0.6390	0.1781	0.3828	0.1550	0.6950
Dog	0.4142	0.2641	0.2821	0.1696	1.2014
Flamingo	0.4967	0.2272	0.5017	0.1595	0.8598
Lucia	0.6699	0.1165	0.5450	0.1181	0.4302
Soccerball	0.5628	0.1271	0.4934	0.1188	0.4642
Train	0.6639	0.1575	0.4372	0.1323	0.5657
Media	**0.5382**	**0.1594**	**0.4094**	**0.1370**	**0.7808**

Tabla 27: Experimento 2: Resultados obtenidos para 9 secuencias de DAVIS seleccionando un umbral de 0.4.

	$\mathcal{J}$		$\mathcal{F}$		$\mathcal{T}$
Secuencias	Mean↑	Std↓	Mean↑	Std↓	Mean↓
Blackswan	0.3651	0.1150	0.3770	0.1396	1.2928
Breakdance	0.2415	0.1464	0.2042	0.1008	1.4374
Bus	0.8031	0.1081	0.4867	0.1344	0.3238
Car-roundabout	0.6725	0.1725	0.4131	0.1605	0.5861
Dog	0.4392	0.2621	0.3123	0.1693	1.2205
Flamingo	0.4481	0.2349	0.4757	0.1698	0.9802
Lucia	0.6617	0.1202	0.5539	0.1199	0.5516
Soccerball	0.5665	0.1442	0.4935	0.1367	0.4569
Train	0.6409	0.1696	0.3966	0.1602	0.7476
Media	**0.5776**	**0.1637**	**0.4326**	**0.1434**	**0.8641**

Tabla 39: Experimento 5: Resultados obtenidos para 9 secuencias de DAVIS sin tasa de olvido